

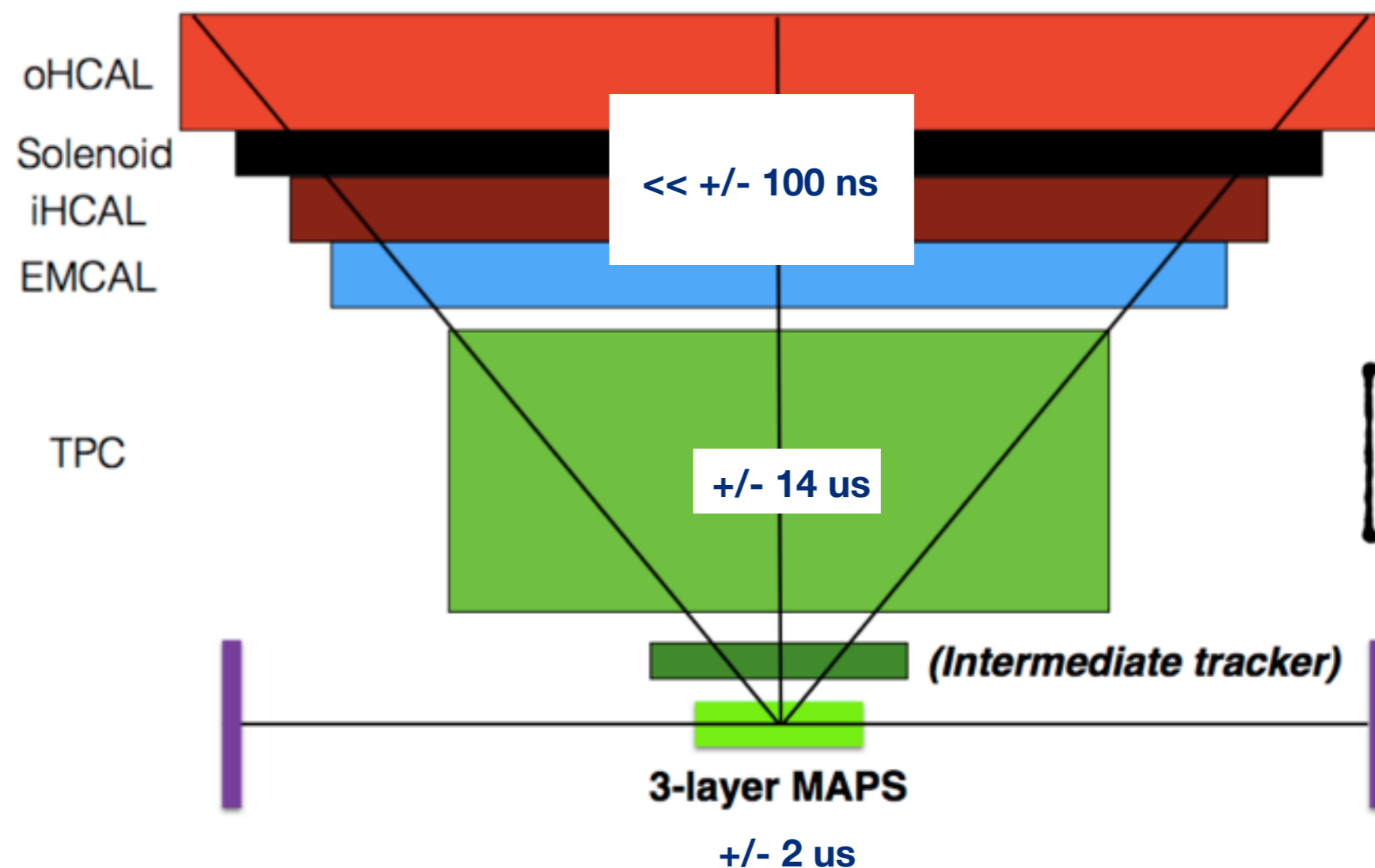
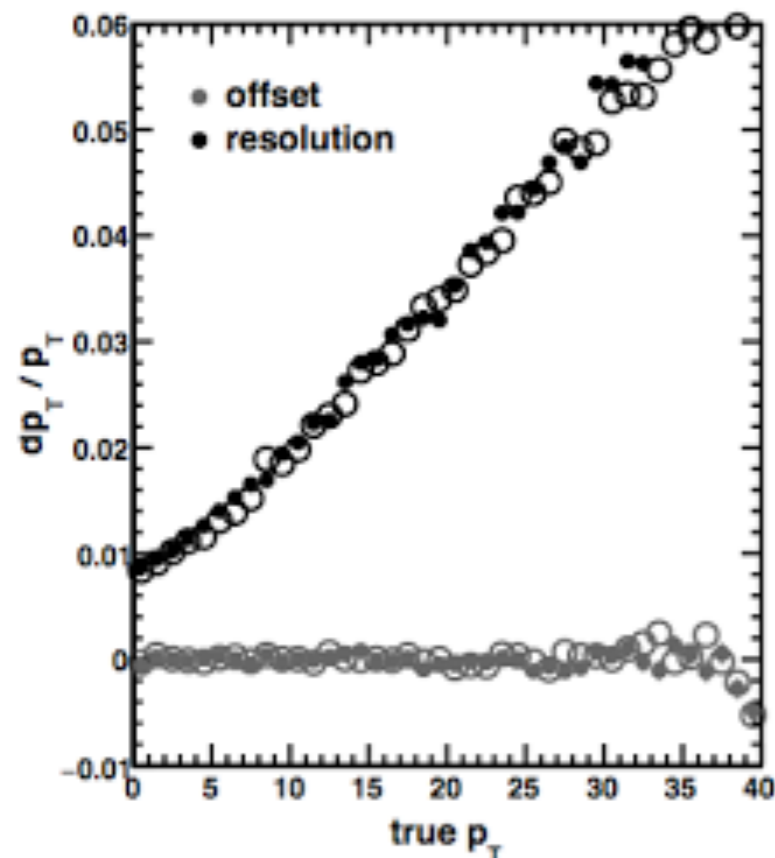
Pileup

1

I've moved on from thinking **space charge** is the tracking system's biggest open question to thinking about **pileup**.

Space Charge (20 cm TPC) - solid points

No Space Charge - open points



What helps: TPC ~ 6 cm / us drift velocity

Key design parameter: z-pitch of intermediate tracker

Pileup Implementation

Instead of fast pileup simulations for each subsystem... **Use GEANT4 time dependence to simulate pileup events**

- (1) Toss pileup events using a poisson distribution, the beam collision rate, and the time between crossings
 - (2) In the conversion of g4hits => g4cells there is already an upper integration limit, modify this to a window
- Off you go...

One event generator for the foreground event...

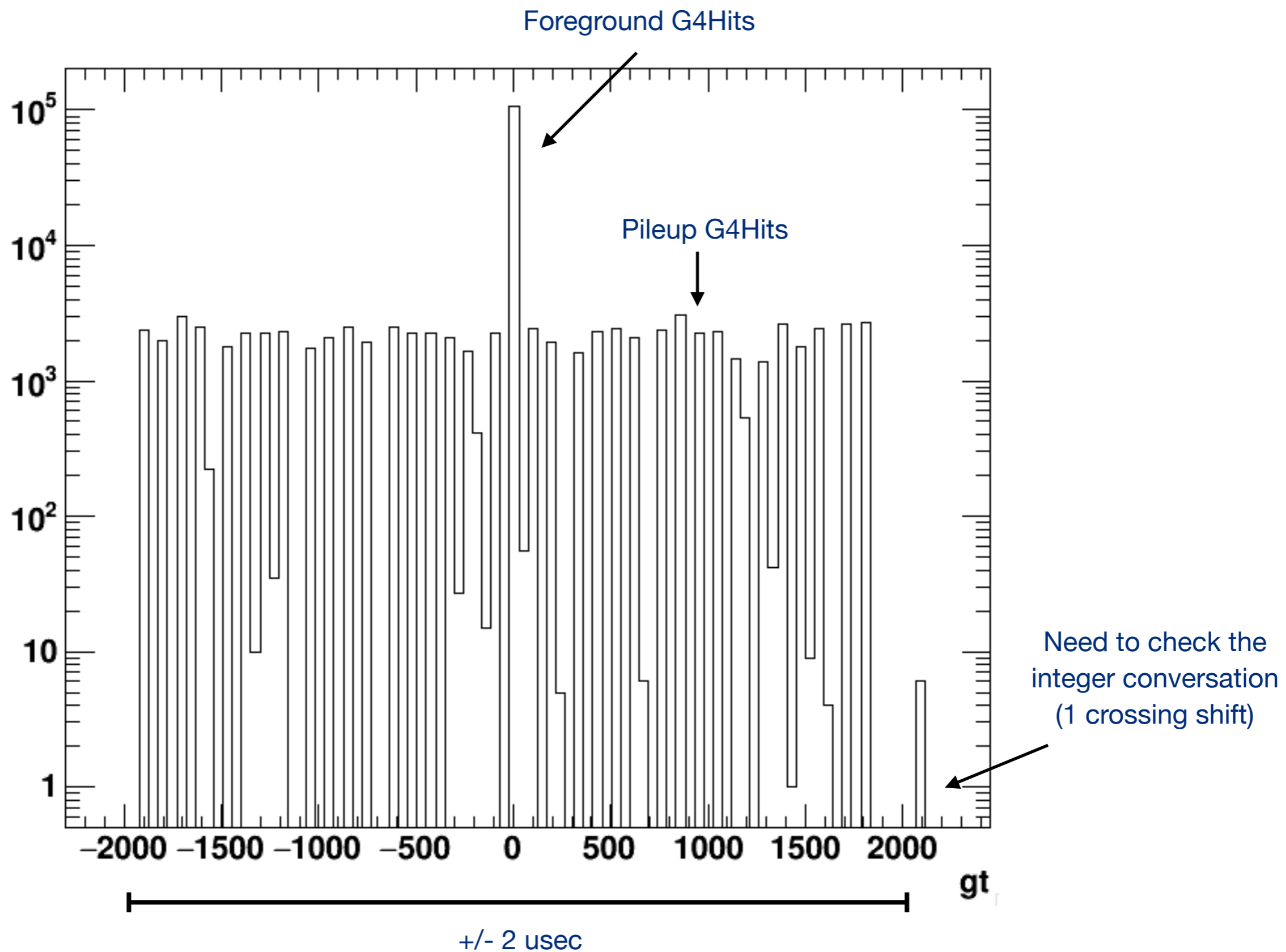
```
PHG4SimpleEventGenerator *gen = new PHG4SimpleEventGenerator();
gen->add_particles("pi-",10); // mu+,e+,proton,pi+,Upsilon
gen->add_particles("pi+",10); // mu-,e-,anti_proton,pi-
if (readhepmc && i==0) {
    gen->set_reuse_existing_vertex(true);
    gen->set_existing_vertex_offset_vector(0.0,0.0,0.0);
} else {
    gen->set_vertex_distribution_function(PHG4SimpleEventGenerator::Gaus,
                                       PHG4SimpleEventGenerator::Gaus,
                                       PHG4SimpleEventGenerator::Gaus);
    gen->set_vertex_distribution_mean(0.0,0.0,0.0);
    gen->set_vertex_distribution_width(0.02,0.02,20.0);
}
gen->set_vertex_size_function(PHG4SimpleEventGenerator::Uniform);
gen->set_vertex_size_parameters(0.0,0.0);
gen->set_eta_range(-1.5, 1.5);
gen->set_phi_range(-1.0*TMath::Pi(), 1.0*TMath::Pi());
gen->set_pt_range(0.1, 40.0);
gen->set_t0(0.0);
gen->Embed(1);
gen->Verbosity(0);
se->registerSubsystem(gen);
```

Another event generator for the pileup events...

```
PHG4SimpleEventGenerator *bkg = new PHG4SimpleEventGenerator();
bkg->add_particles("mu-",4); // mu+,e+,proton,pi+,Upsilon
bkg->add_particles("mu+",4); // mu-,e-,anti_proton,pi-
if (readhepmc && i==0) {
    bkg->set_reuse_existing_vertex(true);
    bkg->set_existing_vertex_offset_vector(0.0,0.0,0.0);
} else {
    bkg->set_vertex_distribution_function(PHG4SimpleEventGenerator::Gaus,
                                       PHG4SimpleEventGenerator::Gaus,
                                       PHG4SimpleEventGenerator::Gaus);
    bkg->set_vertex_distribution_mean(0.0,0.0,0.0);
    bkg->set_vertex_distribution_width(0.02,0.02,20.0);
}
bkg->set_vertex_size_function(PHG4SimpleEventGenerator::Uniform);
bkg->set_vertex_size_parameters(0.0,0.0);
bkg->set_eta_range(-1.5, 1.5);
bkg->set_phi_range(-1.0*TMath::Pi(), 1.0*TMath::Pi());
bkg->set_pt_range(0.1, 40.0);
bkg->Verbosity(0);

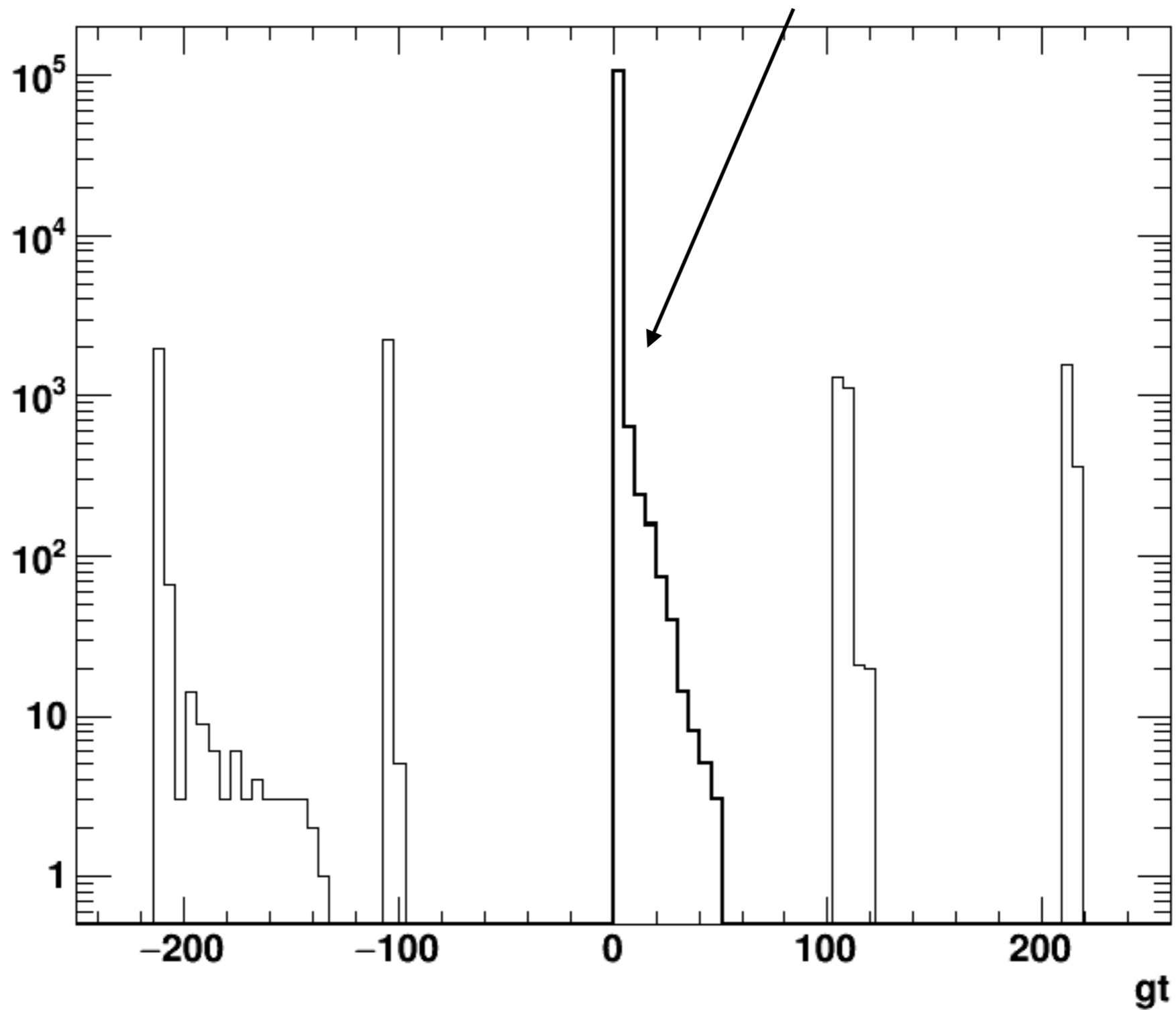
PHG4PileupGenerator *pileup = new PHG4PileupGenerator();
pileup->set_time_between_crossings(106.0);
pileup->set_collision_rate(500.0);
pileup->set_time_window(-2000.0,+2000.0);
pileup->set_generator(bkg);
se->registerSubsystem(pileup);
```

Initial Results



Initial Results (Zoomed)

Foreground G4Hits converted to G4Cells
with a 50 ns window



Some Next Steps

Event Generator Reorganization:

Multiple entry points (HEPMC & InEvent Node)
HEPMC interface not re-run safe

Prevent CPU usage on out of time
hadronic showers

Time dependent Black Hole

Layer Dependent Integration Times

MAPS & TPC will need different windows

